Le protocole RDAP, remplaçant de whois ?

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 26 mars 2015

https://www.bortzmeyer.org/weirds-rdap.html

Traditionnellement, pour trouver de l'information sur un objet enregistré dans une base de données publique de l'Internet (nom de domaine, adresse IP, etc), on utilisait le protocole whois (qui avait été normalisé, longtemps après sa création, dans le RFC 3912 [1]). Ce protocole a de grosses limitations (décrites au paragraphe suivant) et plusieurs tentatives ont déjà été faites pour le remplacer. Le nouveau venu, RDAP ("*Registration Data Access Protocol*"), va-t-il mieux réussir que les précédents ?

Attention, comme le note le document SAC-051 « "*SSAC Report on Domain Name WHOIS Terminology and Structure*" <https://www.icann.org/en/system/files/files/sac-051-en.pdf> », le terme « whois » est souvent employé incorrectement. Il désigne normalement un protocole (celui normalisé dans le RFC 3912) mais est également utilisé pour désigner un service (celui d'accès aux données d'enregistrement) voire pour désigner les données elles-mêmes (« informations WHOIS », terme erroné mais fréquent). Voici un exemple d'utilisation de whois sur Unix, avec le logiciel client GNU whois, pour avoir de l'information sur le domaine `reflets.info` :

```
% whois reflets.info
Domain Name:REFLETS.INFO
Creation Date: 2010-12-23T13:46:11Z
Updated Date: 2014-12-12T17:24:31Z
Sponsoring Registrar:Gandi SAS (R191-LRMS)
...
Registrant Name:Antoine Champagne
Registrant Organization:
Registrant Street: Whois Protege / Obfuscated whois
Registrant Street: Gandi, 63-65 boulevard Massena
Registrant City:Paris
Registrant State/Province:
Registrant Postal Code:75013
Registrant Country:FR
Registrant Phone:+33.170377666
Registrant Email:ea208c9533d64ffbaa6ff82bdbf084d4-3087564@contact.gandi.net
```

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc3912.txt

Parmi les principales limites de whois :

— Aucun mécanisme d'authentification, donc pas de possibilité de restreindre les données selon le client (certains registres utilisent l'adresse IP du client pour donner des privilèges à certains clients),
— Aucun mécanisme de confidentialité (ce point est lié au précédent : comme tout est public actuellement, la confidentialité ne servirait à rien),
— Aucun mécanisme standard pour fournir des options spécifiques à la recherche,
— Aucune structuration des données : le client doit analyser des dizaines de formats différents pour trouver ce qu'il cherche (certains logiciels le font pour lui, comme Net : :DRI <http://search.cpan.org/dist/Net-DRI/> mais ils sont rarement complets, il y a toujours un serveur whois quelque part qui suit des règles différentes et non reconnues) ; même chose pour les messages d'erreur (comme « entité non trouvée »),
— Aucune internationalisation : peut-on envoyer de l'Unicode en réponse à un client et, si oui, avec quel encodage ?
— Aucune façon normalisée d'adapter le résultat au lecteur (par exemple envoyer une adresse en caractères chinois ou bien en caractères latins selon le client),
— Aucun mécanisme standard pour trouver le serveur pertinent pour un objet donné : chaque client whois utilise un truc particulier (par exemple, GNU whois a une liste de serveurs, qu'on peut modifier en éditant un fichier de configuration `/etc/whois.conf`).

Cette liste est connue depuis longtemps. C'est ainsi que le RFC 3707 dressait déjà un cahier des charges d'un bon successeur de whois.

En mai 2012, après pas mal de discussions <https://www.bortzmeyer.org/iris-ou-rest-whois.html>, l'IETF a créé le groupe de travail WEIRDS <https://tools.ietf.org/wg/weirds> pour produire un remplaçant à whois. Ce groupe WEIRDS publie aujourd'hui RDAP (Registration Data Access Protocol), le nouveau protocole. Ses principes ? Modularité (on peut utiliser plusieurs protocoles de transport, le langage d'expression des requêtes est séparé de la définition du format des réponses), et le respect des modes actuelles. Le premier transport normalisé utilise REST et le premier format de réponses est bâti sur JSON (RFC 8259). La requête RDAP avec REST+JSON équivalente à la requête whois citée plus haut serait (cf. la documentation du service RDAP expérimental de .info <http://rdg.afilias.info/rdap/help>) :

```
% curl   http://rdg.afilias.info/rdap/domain/reflets.info
{
   "entities": [
      {
         "links": [
            {
               "href": "http://rdg.afilias.info/rdap/entity/AC12196-GANDI",
               "rel": "self",
               "type": "application/rdap+json",
               "value": "http://rdg.afilias.info/rdap/entity/AC12196-GANDI"
            }
         ],
         "objectClassName": "entity",
         "roles": [
            "technical",
            "billing",
            "administrative",
            "registrant"
         ],
         "vcardArray": [
            "vcard",
            [
               [
                  "version",
                  {},
                  "text",
                  "4.0"
               ],
```

```
            [
               "fn",
               {},
               "text",
               "Antoine Champagne"
            ],
            [
               "adr",
               {},
               "text",
               [
                  "",
                  "",
                  "Whois Protege / Obfuscated whois",
                  "Paris",
                  "",
                  "75013",
                  "FR"
               ]
            ],
            [
               "email",
               {},
               "text",
               "ea208c9533d64ffbaa6ff82bdbf084d4-3087564@contact.gandi.net"
            ],
            [
               "tel",
               {
                  "type": "work"
               },
               "uri",
               "tel:+33.170377666"
            ],
            [
               "tel",
               {
                  "type": "fax"
               },
               "uri",
               "tel:+33.143730576"
            ]
         ]
      ]
...
```


Les RFC qui forment la définition de RDAP sont :

— RFC 7482, « "*Registration Data Access Protocol Query Format*" », explique comment former des requêtes RDAP, sous forme d'URL.
— RFC 7483, « "*JSON Responses for the Registration Data Access Protocol (RDAP)*" », décrit le format des réponses du serveur RDAP, format qui utilise JSON.
— RFC 7480, « "*HTTP usage in the Registration Data Access Protocol (RDAP)*" », normalise l'utilisation de HTTP pour transporter les requêtes et réponses RDAP.
— RFC 7484, « "*Finding the Authoritative Registration Data (RDAP) Service*" », explique comment trouver le bon serveur RDAP.
— RFC 7481, « "*Security Services for the Registration Data Access Protocol*" », détaille les services de sécurité de RDAP (en général fournis par HTTP).
— RFC 7485, « "*Inventory and Analysis of WHOIS Registration Objects*" », qui n'est pas à proprement parler une partie de la définition de RDAP mais qui a servi à spécifier RDAP.

RDAP remplacera-t-il whois ? C'est que whois en a eu, des concurrents malheureux, et que tous sont bien oubliés aujourd'hui. Dans l'ordre rétro-chronologique :

— IRIS (RFC 3981), qui reposait sur XML et un protocole spécifique. Trop compliqué, il a eu peu de déploiements (DENIC avait un service IRIS mais y a renoncé <http://www.denic.de/en/denic-in-dialogue/news/3767.html>).

— LDAP avait été sérieusement proposé comme concurrent mais sans jamais de succès.
— rwhois (RFC 2167), le seul à avoir connu un réel déploiement, à l'ARIN (voir la documentation <http://projects.arin.net/rwhois/>).
— whois++ (RFC 1913).

En tout cas, RDAP est déjà largement mis en œuvre (dix implémentations dont l'interopérabilité a été testée au cours de réunions IETF) même s'il n'y a guère de déploiement en production pour l'instant.

L'exemple plus haut était pour trouver de l'information sur un nom de domaine. Mais RDAP marche aussi pour les adresses IP :

```
%  curl http://rdap.apnic.net/ip/2001:dc7:dd01:0:218:241:97:42
{
  "handle" : "2001:0DC7::/32",
  "startAddress" : "2001:dc7::",
  "endAddress" : "2001:dc7:ffff:ffff:ffff:ffff:ffff:ffff",
  "name" : "CNNIC-CN-20040913",
  "type" : "ALLOCATED PORTABLE",
  "country" : "CN",
  "parentHandle" : "2001:0C00::/23",
  "entities" : [ {
    "handle" : "IPAS1-AP",
    "vcardArray" : [ "vcard", [ [ "version", { }, "text", "4.0" ], [ "fn", { }, "text", "CNNIC IPAS CONFEDER
      "label" : "No.4, Zhongguancun No.4 South Street,\\nHaidian District, Beijing"
    }, "text", [ "", "", "", "", "", "", "" ] ], [ "tel", {
      "type" : "voice"
    }, "text", "+86-010-58813000" ], [ "tel", {
      "type" : "fax"
    }, "text", "+86-010-58813075" ], [ "email", { }, "text", "ipas@cnnic.cn" ] ] ],
...
```

C'est bien compliqué, tout ce JSON. Et si on veut juste extraire certaines informations ? On peut utiliser un client RDAP spécialisé, ou écrire soi-même un traitement, ou encore se servir des processeurs JSON tout faits comme jq <https://www.bortzmeyer.org/jq.html> :

```
%  curl -s http://rdap.apnic.net/ip/2001:dc7:dd01:0:218:241:97:42 | jq .country
"CN"
```

Questions mises en œuvre, on peut citer :

— Le serveur <https://github.com/DNSBelgium/rdap> écrit en Java,
— Le client nicinfo <https://github.com/arineng/nicinfo> (en Ruby),
— Le client rdapper <http://search.cpan.org/~gbrown/rdapper-0.08/> (en Perl).
— Et, comme dans les exemples plus haut, n'importe quel client HTTP...

De bonnes lectures sur RDAP et le travail du groupe WEIRDS :

— Un résumé par un des auteurs du protocole <http://www.circleid.com/posts/20150121_where_do_old_protocols_go_to_die/>.
— Un bon résumé par Marc Blanchet <http://singapore52.icann.org/en/schedule/mon-tech/presentation-ietf-09feb15-en.pdf>.

Et merci au dit Marc Blanchet pour sa relecture attentive de mon article.